

What is claimed is:

1. An apparatus for utilizing information, comprising:
a memory, the memory comprising at least one data structure; and
5 a plurality of layers, each layer comprising at least one thread, each thread utilizing each data structure from the same portion of the memory.
2. The apparatus of claim 1, further comprising an application layer and a hardware layer, wherein the application layer comprises one of the plurality of layers, wherein the hardware layer comprises one of the plurality of layers,
10 wherein the application layer and hardware layer utilize each data structure from the same portion of memory.
3. The apparatus of claim 2 wherein at least one of the plurality of layers comprises a realtime thread.
4. The apparatus of claim 1, wherein each data structure comprises a block object,
15 wherein at least a portion of each block object is comprised of a contiguous portion of the memory.
5. The apparatus of claim 4, wherein the contiguous portion of the memory is defined a byte array.
6. The apparatus of claim 1, wherein the at least one data structure comprises a block object.
20
7. The apparatus of claim 1, further comprising a Java or Java-like virtual machine, wherein each thread comprises a Java or Java-like thread, wherein the Java or Java-like thread utilizes the same portion of memory independent of Java or Java-like monitors.
- 25 8. The apparatus of claim 1, the apparatus further comprising interrupt means for disabling interrupts; and a Java or Java-like virtual machine capable of executing each thread, wherein each thread utilizes the same portion of memory after the interrupts are disabled by the interrupt means.
9. The apparatus of claim 8, wherein all interrupts are disabled before each thread
30 utilizes the same portion of memory.
10. The apparatus of claim 8, wherein the threads disable the interrupts via the interrupt means.

11. The apparatus of claim 1, wherein the information is received by the apparatus as
streamed information, wherein each data structure is preallocated to the memory
prior reception of the information.
12. The apparatus of claim 4, further comprising a freelist data structure, wherein each
5 block object is preallocated to the freelist data structure by the apparatus prior to
utilization of the information.
13. The apparatus of claim 12, the apparatus further comprising a protocol stack, the
protocol stack residing in the memory, wherein the protocol stack preallocates each
block to the freelist data structure.
- 10 14. The apparatus of claim 1, the apparatus further comprising a virtual machine, the
virtual machine utilizing a garbage collection mechanism, the virtual machine
running each thread, each thread utilizing the same portion of the memory
independent of the garbage collection mechanism.
- 15 15. The apparatus of claim 14, wherein the garbage collection mechanism comprises a
thread, wherein the threads comprise Java-like threads, wherein the threads each
comprise a priority, wherein the priority of the Java-like threads is higher than the
priority of the garbage collection thread.
16. The apparatus of claim 1, wherein each data structure comprises a block object,
and further comprising a freelist data structure and at least one queue data
20 structure, each block object comprising a respective handle, wherein at any given
time the respective handle belongs to the freelist data structure or a queue data
structure.
17. The apparatus of claim 6, further comprising, at least one queue data structure; and
at least one frame data structure, each frame data structure comprising an instance
25 of one or more block objects, each block object comprising a respective handle,
each queue data structure capable of holding an instance of at least one frame data
structure, and each thread using the queue data structure to pass a block handle to
another thread.
18. The apparatus of claim 1, further comprising a virtual machine, the virtual machine
30 running each thread; at least one queueendpoint, each queueendpoint comprising at
least one of the threads; and at least one queue, each queue comprising ends, each
end bounded by a queueendpoint, each queue for holding each of data structures in

- a data path for use by each queueendpoint, wherein each queue notifies a respective queueendpoint when the queue needs to be serviced by the queueendpoint, wherein a queueendpoint passes instances of each data structure from one queue to another queue by a respective handle belonging to the data structure.
- 5 19. The apparatus of claim 18, wherein a queue notifies a respective queueendpoint upon the occurrence of a queue empty event, queue not empty event, queue congested event, or queue not congested event.
20. The apparatus of claim 19, further comprising a queue status data structure shared by a queue and a respective queueendpoint, wherein the queue sets a flag in
10 the data status structure to notify the respective queueendpoint when the queue needs to be serviced.
21. An apparatus for utilizing a stream of information in a data path, comprising:
a memory, the memory comprising at least one data structure, each data structure comprising a pointer;
15 a plurality of layers, the data path comprising the plurality of layers, the stream of information comprising the at least one data structure, each layer utilizing each data structure via its pointer.
22. The apparatus of claim 21, wherein each layer comprises at least one thread, each thread utilizing each data structure from the same portion of the memory.
- 20 23. The apparatus of claim 22, further comprising an interrupt disabling mechanism; and at least one queue, each queue disposed in the data path between a first layer and a second layer, the first layer comprising a producer thread, the second layer comprising a consumer thread, the producer thread for enqueueing each data structure onto a queue, the consumer thread for dequeing each data structure from
25 the queue, wherein prior to dequeing and enqueueing each data structure interrupts are disabled.
24. The apparatus of claim 22, the apparatus further comprising a virtual machine, the virtual machine comprising a garbage collection mechanism, the virtual machine running each thread independent of the garbage collection mechanism.
- 30 25. A system for utilizing data structure with a plurality of threads, comprising:
an interrupt mechanism for enabling and disabling interrupts;
a memory, the memory comprising at least one data structure; and

a plurality of threads, the plurality of threads utilizing the data structures after disabling interrupts with the interrupt mechanism.

26. The system of claim 25, wherein the plurality of threads utilize each of the data structures from the same portion of memory.

5 27. A system for accessing streaming information with a plurality of threads, comprising:

a memory; and

10 interrupt means for enabling and disabling interrupts; wherein the plurality of threads access the streaming information from the memory by disabling the interrupts via the interrupt means.

28. The system of claim 27, further comprising a memory, wherein the plurality of threads access the streaming information from the same portion of the memory.

29. A method for accessing information in a memory with a plurality of threads, comprising the steps of:

15 transferring information from one thread to another thread via handles to the information; and

disabling interrupts via the threads before performing the step of transferring the information.

20 30. The method of claim 29, further comprising a step of accessing the information with the plurality of threads from the same portion of the memory.